Днес Python е един от най-използваните програмни езици и един от най-предпочитаните от програмистите. Той има интуитивен и разбираем синтаксис и върви с редица рамки, които да подсилят работата ти.

Винаги е била повсеместна дилема, когато става въпрос за избора на рамка за следващия ви проект. На всеки няколко месеца виждате нова технология и рамка, която преодолява слабостта на предишната, която сте използвали.

В мемента двете най използвани и популярни Python рамки са **Flask** и **Django** като и двете си има собствените си минуси и плюсове.

Но без значение коя от двете рамики ще изберете няма да сгрешите в никакъв случай

Първоначално стартиран през 2010 г. от Armin Ronacher, Flask е известен като микро-рамка, тъй като има малка или никаква зависимост от външни библиотеки. Използвайки го, разработчиците имат гъвкавостта да избират шаблони за проектиране, инструменти и бази данни. По този начин гъвкавостта е основната характеристика на тази Python рамка.

Тъй като индустрията за уеб разработка сега клони повече към микро-услуги и server-less платформи, популярността на Flask непрекъснато нараства. Той се използва широко за изграждане на мащабируеми уеб приложения без усилие. Всичко това благодарение на неговите уникални характеристики.

### **Features of Flask**

* Вграден сървър за разработка и бърз дебъгер
* Висока мащабируемост
* Използва шаблони Jinja2 и е съвместим с WSGI 1.0
* ORM-agnostic and highly flexible
* HTTP рекуести които подържат функционалноста
* Дава възможност за unit testing
* Лесен за работа

КАКВО Е Django

Стартиран през 2005 г., Django е рамка с пълен стек, проектирана с основната цел да улесни разработването на сложни уебсайтове, управлявани от база данни. Той улеснява сигурното и по-бързо разработване, като освобождава разработчиците от извършването на повтарящи се задачи за уеб разработка. С много отлични функции, той позволява на разработчиците да създават стабилни и високоефективни приложения.

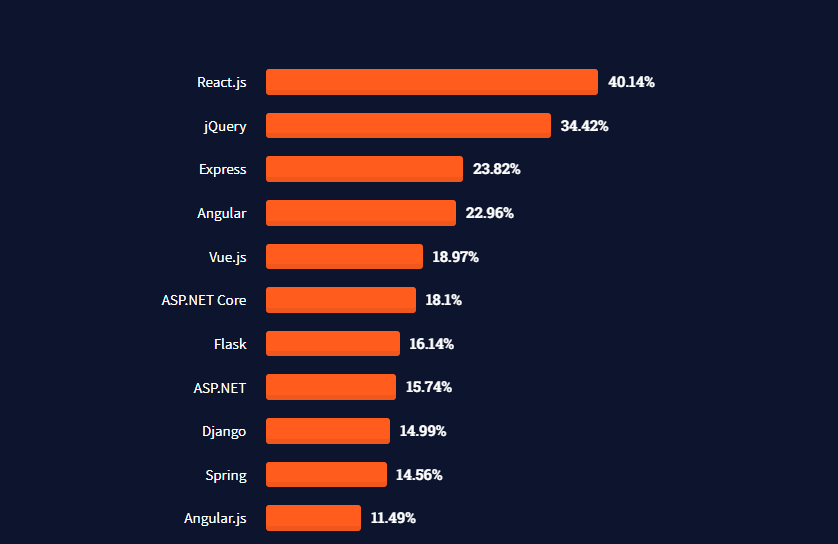
### **Features of Django**

* Позволява бързо разработка
* лесен за мащабиране и подържане
* Много сигурен и многофункционален
* Осигурява поддръжка за много front-end инструменти

Докато Flask и Django са рамки на Python, подходящи за разработване на уеб приложения, които са с висока производителност и лесни за мащабиране, те са много различни една от друга.

Макар Django да позволява бързо създаване на приложения той е сравнително по-бавен от Flask.

## Мирко-рамка Flask

Flask е т.нар. **micro-framework**, много лека микро-рамка, която контрастира с full-stack рамки като Django. Основната ѝ функционалност е малка, но е чувствително разширяема, правейки я подходяща за разработка както на малки, така и на големи приложения. В момента е една от най-използваните технологични рамки изобщо, по данни от анкетата на StackOverflow, и е в топ 3 на най-използваните рамки с Python. 

Едно от големите предимства на Flask е фактът, че е много **интуитивна рамка** и е лесна за използване, както от по-неопитни програмисти, така и от експерти. Известно е, че популярни компании като Netflix, LinkedIn, AirBnB и reddit използват framework-а в работата си.

Въпреки че е подходяща за начинаещи, Flask предлага всички функционалности, необходими за създаването на **back-end-а на добре работещо уеб приложение**. Тя идва с технологиите и инструментите, с които да създаваш необходимите ти функционалности, фокусирайки се върху тях, а не върху дизайна и визията на приложението.

Чрез нея можеш бързо и лесно да започнеш да разработваш уеб апликации, които могат да бъдат **скалирани**. Проектът ти, разработен с помощта на Flask, ще има изчистен и подреден код, без парчета за функционалности, които не се ползват и не са необходими на този етап.

Предвид това колко широкоразпространено е ползването на рамката, можеш да очакваш да попаднеш в **богата и активна общност** от разработчици. Именно на тях и базата от знания, които са развили и придобили, можеш да разчиташ, ако срещаш трудности или имаш въпроси за работата с Flask.

Макар и лека микро-рамка, Flask ти помага да разработваш както простички, така и мащабни **приложения с разнообразни цели**. С него можеш да създадеш блог, приложения като социални мрежи, метеорологични апликации, форми за събиране на обратна връзка, сайт за портфолио, с който да демонстрираш на потенциални работодатели, че имаш опит с рамката, а дори и да разработиш machine learning модели.

Когато имаш нужда от **гъвкавост** в проектите си и търсиш **възможности за персонализиране**, то рамката няма да те подведе. Особено ако става дума за **технологични решения**, които ще се отложат във времето, тези опции за персонализиране и промяна ще ти бъдат от голяма полза. Това по свой начин може да подпомогне производителността ти, така че **да ускориш процеса** по разработка.

Именно в това се крие и основното предимство пред цялостни, комплексни рамки като Django. Ще можеш бързо да създаваш уеб приложения, към които да добавяш нови и наистина необходими функционалности **на по-късен етап**.